10/643-079 - 12.08.03

(12) UK Patent Application (19) GB (11) 2 324 179 (13) A

(43) Date of A Publication 14.10.1998

(21) Application No 9802087.8

(22) Date of Filing 30.01.1998

(30) Priority Data
(31) 970262 (32) 10.04.1997 (33) IE

(71) Applicant(s)
Stampalia Limited
(Incorporated in Ireland)
Priority House, 63 Patrick Street, Dun Laoghaire,
County Dublin, Ireland

(72) Inventor(s)
Alexander Thomas Florence
Alan Joseph Gorman

(74) Agent and/or Address for Service
G F Redfern & Co
7 Staple Inn, Holborn, LONDON, WC1V 7QF,
United Kingdom

(51) INT CL[6]
G06F 1/00 12/14

(52) UK CL (Edition P)
G4A AAP

(56) Documents Cited
US 5012514 A

(58) Field of Search
UK CL (Edition P) G4A AAP
INT CL[6] G06F

(54) Abstract Title
Preventing access to a hard disc in a computer on booting-up from a floppy disc

(57) A method for preventing access to a C drive hard disc in an IBM compatible PC when the PC has been booted-up from a floppy disc comprises transferring the partition code and partition table from the standard partition sector (1) to an alternative partition sector (3) on the hard disc, in encrypted form. The BPB data and the boot code are encrypted and transferred from the standard boot sector (2) to an alternative boot sector (4) on the hard disc. The standard boot sector is left blank, and an executable protection code is written in the standard partition sector (1). The protection code is read only by the basic operating system of the PC prior to booting-up from the hard disc. The protection code contains an instruction for inserting a protection handler of the protection code in the BIOS interrupt chain as a handler for interrupts for access to the standard partition and boot sectors (1) and (2), and an instruction for directing all valid interrupts for access to the standard partition and boot sectors (1) and (2) to the alternative partition and boot sectors (3) and (4).

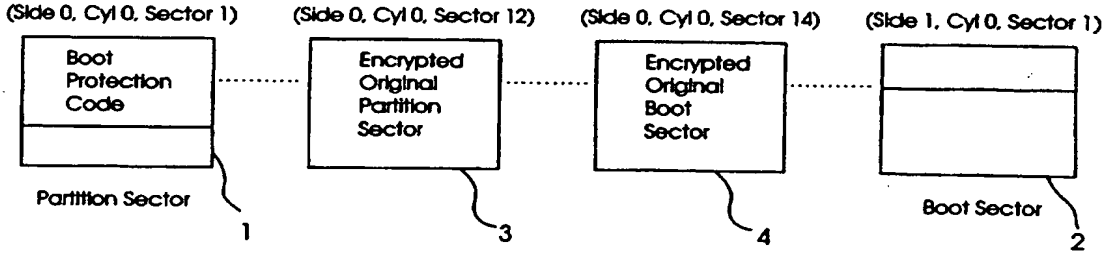

Fig. 2

GB 2 324 179 A

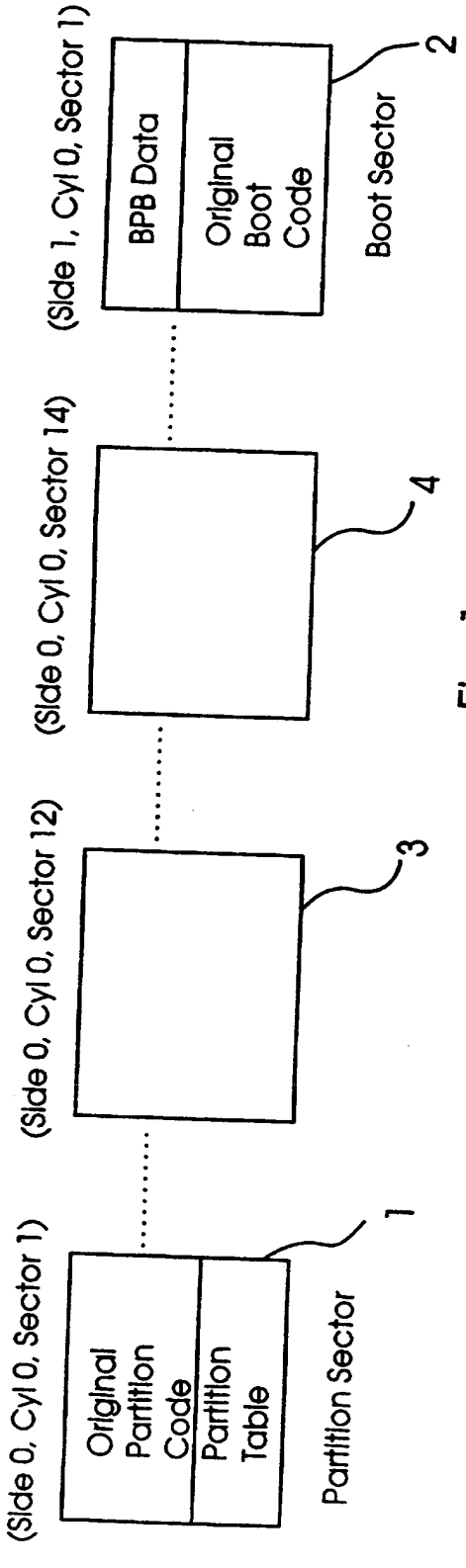

Fig. 1

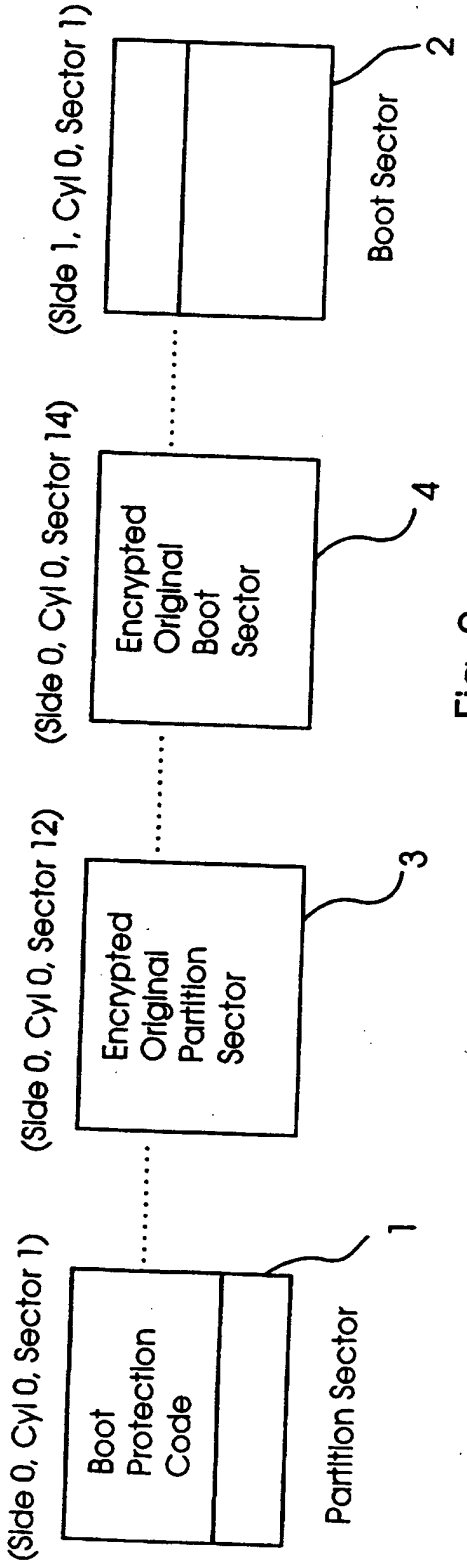

Fig. 2

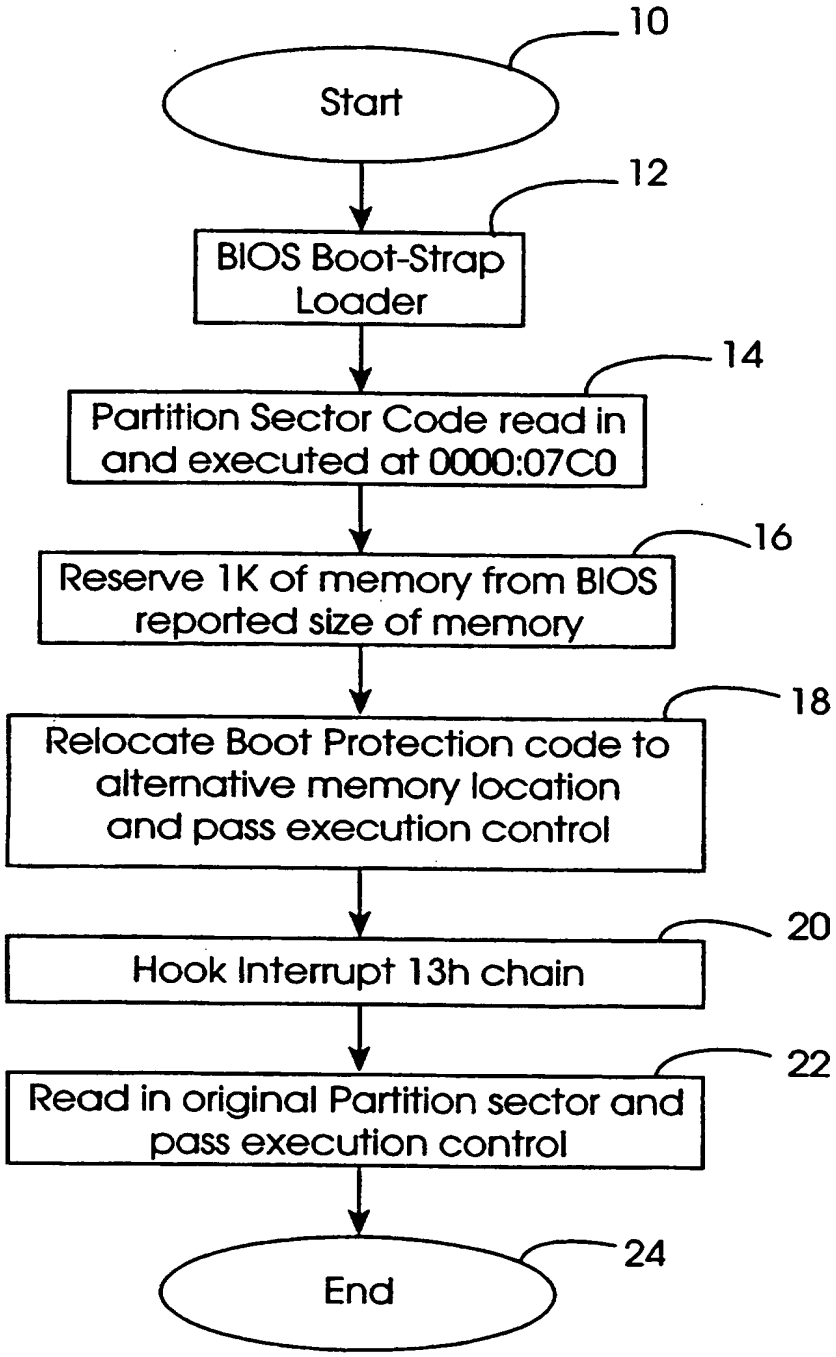

Fig. 3

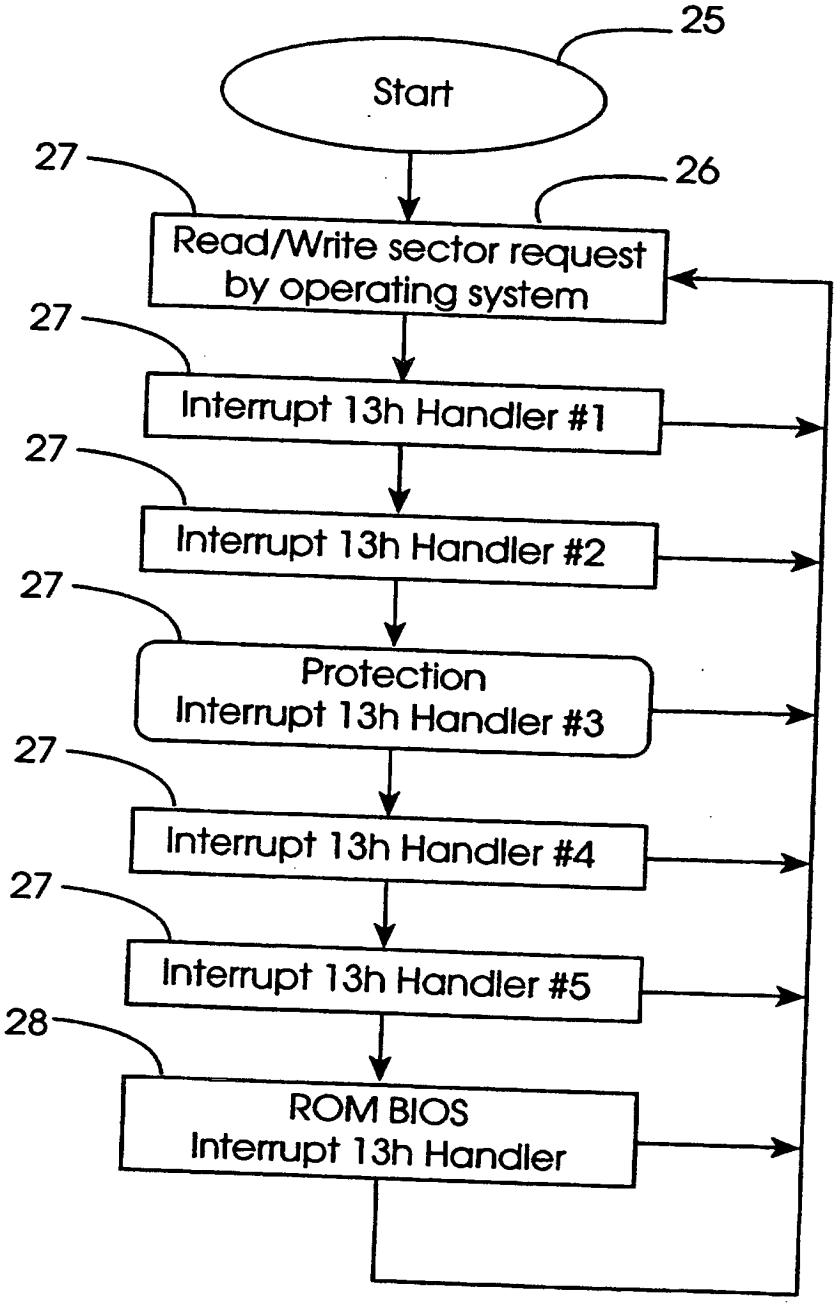

Fig. 4

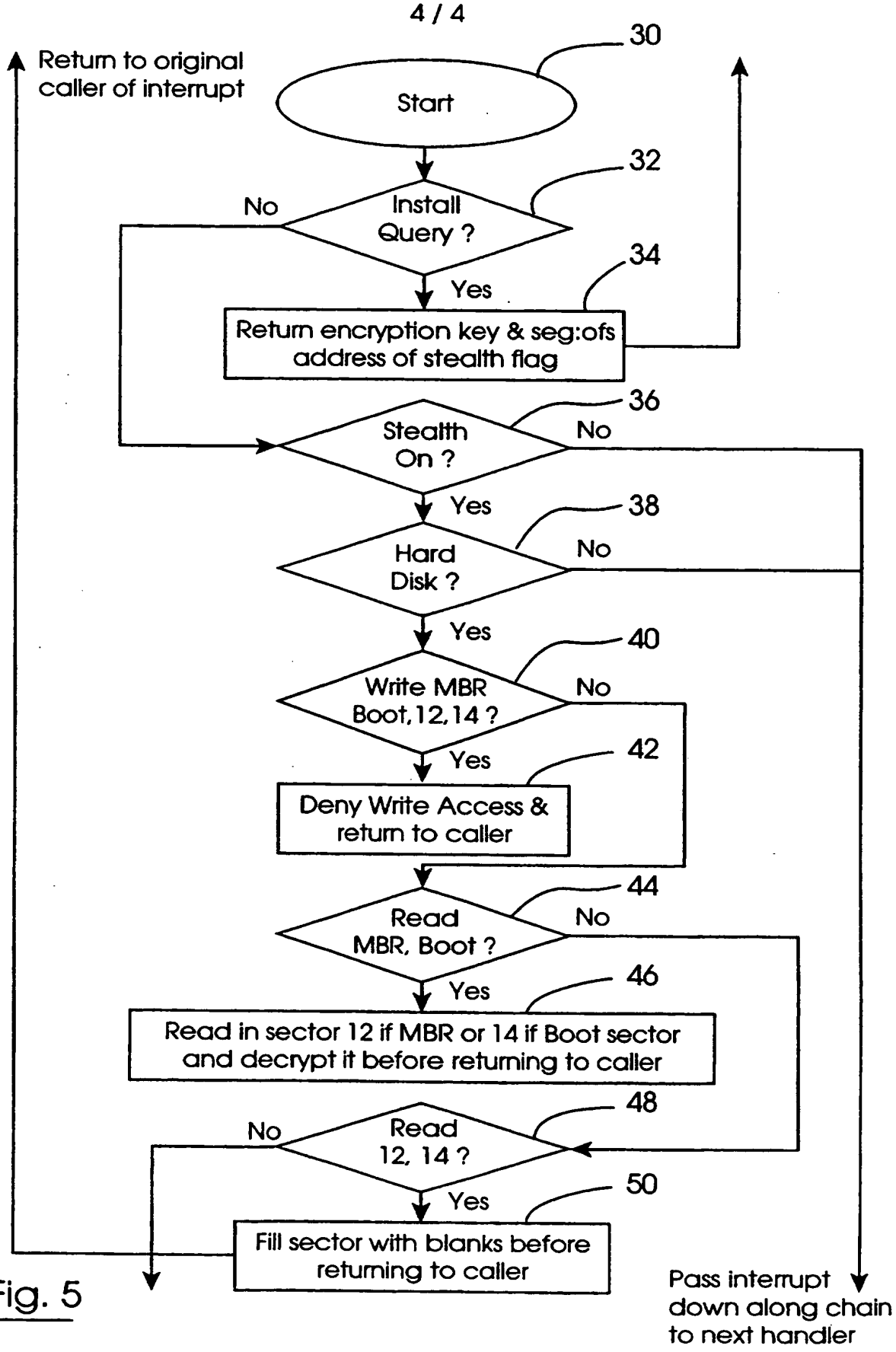

Fig. 5

"A computer and a method for preventing access to a hard disc in a computer on booting-up from a floppy disc"

The present invention relates to a computer and a method for preventing access to a hard disc in a computer when the computer has been booted-up from a floppy disc, and in particular, the invention relates to a method for preventing access to the C drive hard disc of an IBM compatible personal computer (PC) after the PC has been booted-up from a floppy disc in the A or B drives.

In general, it is a relatively straightforward matter to secure data and files stored on a hard disc of a PC by inserting a password protection programme which prevents access to data and files on a hard disc beyond a certain level unless a specific password or passwords are entered into the computer. Such password protection will be well known to those skilled in the art. However, in general, such password protection programmes can readily easily be bypassed by booting-up the computer from a system disc in a floppy disc drive. In the case of an IBM compatible PC, in general, there are two floppy disc drive locations provided which are referred to respectively as the A and B drives of the comput r. The first hard disc of an IBM compatible PC

is called a C driv . If the PC is provided with other hard discs, the remaining hard discs are sequentially identified as D, E, F, etc. drives.

In an IBM compatible computer, when the computer is switched on the basic input output system (BIOS) of the computer sequentially interrogates the respective drives in the order, the A drive first, the B drive second, and the C drive third. If a system floppy disc is inserted in either of the A or B drives, booting-up is carried out from the floppy disc. If a floppy disc is provided in neither of the A and B drives, the BIOS boots-up the computer from the C drive. In booting-up from the C drive, the BIOS initially reads a code which is referred to as the partition code and which is located in a standard location, generally referred to as the standard partition sector. In an IBM compatible PC the standard partition sector is located on the C drive hard disc at position side zero, cylinder zero, sector one. The computer executes the partition code prior to booting-up. The computer then reads the partition table in the standard partition sector and proceeds to the boot-up sector which is located at a standard location, generally referred to as the standard boot sector. In an IBM compatible PC the standard boot sector is at side one, cylinder zero, sector one. Booting-up then commences, and after

booting-up if password protection is provided the user is interrogated for a password. In the absence of a valid password the computer fails to proceed further. On the other hand, should a system disc be located in the A or B drive, the computer boots-up from the system disc, and bypasses the password protection. On boot-up being completed, the operating system then reads the partition table which is located in the standard partition sector of the hard disc. In this way, the password protection is bypassed.

There is therefore a need for a method for preventing access to a hard disc in a computer when the computer has been booted-up from a system disc in a floppy disc drive of the computer, and there is also a need for a computer so protected.

The present invention is directed towards providing such a method and a computer.

According to the invention there is provided a method for preventing access to a hard disc in a computer when the computer has been booted-up from a floppy disc, the hard disc being of the type which would normally have an executable partition code and a partition table at a standard location known as the standard partition sector, the executable partition code being read only

prior to booting-up of the computer when the computer is being booted-up from the hard disc, the method comprising the steps of:

transferring the partition code and the partition table to an alternative partition sector on the hard disc, and

writing an executable protection code in the standard partition sector to be read prior to booting-up of the computer when the computer is being booted-up from the hard disc, the protection code comprising;

an instruction for inserting the protection code in the BIOS interrupt chain as a handler for interrupts for access to the standard partition sector, and

an instruction for directing all valid interrupts for access to the standard partition sector of the hard disc to the alternative partition sector.

The method according to the invention makes use of the fact that when a computer has been booted-up from a system disc in a floppy disc drive the basic operating system of the computer does not require the partition code in the standard partition sector, and accordingly, the basic operating system after booting-up from the floppy disc drive is directed to the standard partition sector on the hard disc to read the partition table only. Since th partition table has been transferred to an alternative partition sector on the hard disc,

and since the protection code is provided in the standard partition sector of the hard disc, the basic operating system of the computer, on not finding a partition table code assumes that a hard disc is not installed in the computer, and returns a message to this effect to the user. However, since on booting-up from a hard disc, the basic operating system of the computer first reads the partition code in the standard partition sector, the operating system on reading the executable protection code in the standard partition sector of the hard disc merely proceeds to execute this, thereby permitting booting-up to proceed from the hard disc. In this way, should password protection be installed after booting-up, the user is interrogated for a password.

In one embodiment of the invention the protection code is inserted in the BIOS interrupt chain just before the BIOS handler.

Preferably, the partition code and the partition table are encrypted in the alternative partition sector, and the protection code includes a decryption key for decrypting the partition code and the partition table.

In another embodiment of the invention the boot code and the BPB data code are transferred from a standard

boot sector of the hard disc to an alternative boot sector on the hard disc, and the protection code comprises an instruction for directing all valid interrupts for access to the standard boot sector of the hard disc to the alternative boot sector. Preferably, the boot code and the BPB data code is encrypted in the alternative boot sector.

In another embodiment of the invention the protection code comprises an instruction for inserting the protection code in memory in the computer prior to inserting the protection handler of the protection code in the BIOS interrupt chain.

In a further embodiment of the invention the protection code comprises an instruction for checking if a read interrupt is received for reading either of the alternative partition or boot sectors, and an instruction to return to the caller a predetermined message unrelated to the code stored in the respective alternative sectors should such a read interrupt be received.

Preferably, the protection code comprises an instruction for checking if a write interrupt is rec ived for writing to either of the alternative partition or boot sectors, and an instruction to return

to the caller a message confirming that the respective alternative sectors cannot be written to should such a write instruction be received.

Advantageously, the protection code contains the identity of the location of each of the alternative sectors.

In one embodiment of the invention the method is adapted for preventing access to a hard disc in a personal computer.

In another embodiment of the invention the method is adapted for use with a computer which is an IBM compatible computer, and the method is for preventing access to the C drive hard disc of the computer.

Additionally, the invention provides a computer operating under the control of the method according to the invention for preventing access to a hard disc in the computer after the computer has been booted-up from a floppy disc.

Further, the invention comprises a computer comprising a hard disc, the hard disc being of the type which would normally have an xecutable partition code and an partition table at a standard location known as the

standard partition sector, the executable partition code being read only prior to booting-up of the computer when the computer is being booted-up from the hard disc, the hard disc having written thereon at an alternative partition sector the partition code and the partition table, and an executable protection code being written in the standard partition sector to be read prior to booting-up of the computer when the computer is being booted-up from the hard disc, the protection code comprising:

an instruction for inserting the protection code in the BIOS interrupt chain as a handler for interrupts for access to the standard partition sector, and

an instruction for directing all valid interrupts for access to the standard partition sector of the hard disc to the alternative partition sector.

In one embodiment of the invention the instruction for inserting the protection code in the BIOS interrupt chain as a handler for interrupts for access to the standard partition sector is an instruction for inserting the protection code in the BIOS interrupt chain just before the BIOS handler.

In one embodiment of the invention the partition code and the partition table are encrypted in the alternative partition sector, and the protection code

includes a decryption key for decrypting the partition code and the partition table.

In a further embodiment of the invention the boot code and the BPB data code are written in an alternative boot sector on the hard disc which is different to a standard boot sector in which the boot code and the BPB data code are normally written, and the protection code comprises an instruction for directing all valid interrupts for access to the standard boot sector of the hard disc to the alternative boot sector. Preferably, the boot code and the BPB data code is encrypted in the alternative boot sector.

In one embodiment of the invention the protection code comprises an instruction for inserting the protection code in memory in the computer prior to inserting the protection handler of the protection code in the BIOS interrupt chain.

In another embodiment of the invention the protection code comprises an instruction for checking if a read interrupt is received for reading either of the alternative partition or boot sectors, and an instruction to return to the caller a predetermined message unrelated to the code stored in the respective alternative sectors should such a read interrupt be

received.

In a further embodiment of the invention the protection code comprises an instruction for checking if a write interrupt is received for writing to either of the alternative partition or boot sectors, and an instruction to return to the caller a message confirming that the respective alternative sectors cannot be written to should such a write instruction be received.

Preferably, the protection code contains the identity of the location of each of the alternative sectors.

In one embodiment of the invention the computer is a personal computer.

In a further embodiment of the invention the computer is an IBM compatible personal computer, and the hard disc is the C drive hard disc of the computer.

The invention will be more clearly understood from the following description of a preferred embodiment thereof which is given by way of example only with reference to the accompanying drawings, in which:

Fig. 1 is a diagrammatic representation of four

sectors of a conventional hard disc,

Fig. 2 is a diagrammatic representation of the four sectors of the hard disc of Fig. 1 after having been altered by the method according to the invention,

Fig. 3 is a flow chart of a routine of the method according to the invention which is carried out by the computer when the computer is being booted-up from the hard disc of Fig. 2,

Fig. 4 is a flow chart illustrating a BIOS interrupt chain of the computer, and

Fig. 5 is a flow chart of a routine which is carried out by the computer operating under the method according to the invention.

Referring to the drawings and initially to Figs. 1 and 2, there is illustrated four sectors of a hard disc, namely, a standard partition sector 1, a standard boot sector 2, an alternative partition sector 3 and an alternative boot sector 4. The four sectors 1 to 4 are relevant to the method according to the invention for preventing access to the hard disc when the computer has been booted-up from a system disc in a floppy disc

drive of the computer. In this embodim nt of the invention the computer is an IBM compatible PC, and Fig. 1 illustrates four sectors on the conventional C drive hard disc before the application of the method. The standard partition sector 1 is located at side zero, cylinder zero, sector one, and comprises executable partition code which is read and executed by the basic operating system of the computer prior to booting-up. The standard partition sector 1 also comprises the partition table which lays out the geometry of the hard disc. The code which is stored on the standard partition sector 1 of a C drive hard disc of an IBM compatible PC will be well known to those skilled in the art. The standard boot sector 2 of the C drive hard disc is at location side one, cylinder zero, sector one. The standard boot sector 2 contains the basic input output system (BIOS) parameter block (BPB) data and the boot code which is executed by the computer on booting-up from the hard disc. This will also be well known to those skilled in the art. The alternative partition and boot sectors 3 and 4, respectively, may be any two sectors, but in the present case the alternative partition sector 3 is located at side zero, cylinder zero, sector twelve and the alternative boot sector 4 is located at side zero, cylinder zero, sector fourte n.

Turning now to Fig. 2 the C drive hard disc is illustrated after being altered according to the method of the invention. The method of the invention requires that the partition code and the partition table be transferred from the standard partition sector 1 to the alternative partition sector 3. In accordance with the method of the invention the BPB data and the boot code are also transferred from the standard boot sector 2 to an alternative boot sector 4. In addition to transferring the partition code and partition table and the BPB data and the boot code to the alternative partition Sector 3 and the alternative boot sector 3 and 4, respectively, the partition code and partition table, and the BPB data and the boot code are all encrypted in the respective alternative sectors 3 and 4. The standard boot sector is left completely blank by filling it with zeros.

The next step in the method according to the invention is to write an executable protection code in the standard partition sector 1 which is read and executed by the basic operating system of the computer prior to booting-up from the hard disc. It should be emphasised here that all the partition code and the partition table is entirely removed from the standard partition sector 1 of the hard disc. The protection code written in the standard partition sector 1 contains the

addresses of the locations of the alternative partition sector 3 and the alternative boot sector 4 so that a valid interrupt for either of the standard sectors 1 and 2 is directed to the appropriate alternative sector 3 and 4. Additionally, the protection code comprises a decryption key of one byte for decrypting the partition code, the partition table, the BPB data code and the boot code in the respective alternative partition and boot sectors 3 and 4, respectively. Accordingly, on a valid interrupt being received for access to the standard partition sector 1 or the standard boot sector 2, the interrupt is directed to the appropriate alternative sector 3 or 4, and the relevant information is retrieved and decrypted by the BIOS under the control of the protection code.

Referring now to Fig. 3 a flow chart of a routine which is executed by the basic operating system of the computer when operating under the control of the protection code prior to booting-up will now be described. Block 10 starts the routine, and the routine moves to block 12. Block 12 hands control of the computer to the BIOS which performs the appropriate checks and functions which will be well known to those skilled in the art. The routine then moves to block 14 which reads the protection code from the standard partition sector 1 and loads the protection code into

memory location 0000:07C0, and executes the protection code at this location. The routine then moves to block 16 which seizes one Kbyte of RAM memory from a variable set up previously by the BIOS. The routine then moves to block 18 which relocates the protection code from its initial memory location to the seized RAM memory. The variable set up by the BIOS advises the basic operating system as to the amount of memory installed in the computer. By decrementing this value, the system ignores the boot protection code lying in the seized RAM memory, and so its handler remains resident. The variable is found in the BIOS data area at memory location 0040:0013. The routine then moves to block 20 which hooks the interrupt handler of the protection code into the BIOS interrupt chain, which in an IBM compatible PC is the interrupt 13h I/O chain. The protection interrupt handler is located in the BIOS interrupt chain adjacent the BIOS handler. This is described below with reference to Fig. 4 where the protection interrupt handler is handler No. 3 in the BIOS interrupt chain. In normal operation when an interrupt call is made by the operating system or any application, the interrupt vector table which is found at location 0000:0000 is interrogated to find the vector or address of the interrupt. In this particular case the address of the interrupt is stor d at location 0000:004C. This is because each address takes up four

bytes of storage, thus,

13h (Interrupt) * 4 (Bytes per address)
= 4Ch (Offset into Vector Table)

This address is stored inside the protection interrupt handler No. 3 so that any calls which are not relevant to the protection interrupt handler made to the BIOS interrupt chain can be passed down to the next interrupt handler. This is described in more detail below. The full address of the protection interrupt handler is inserted into the interrupt vector table so that the protection interrupt handler is informed of any subsequent interrupts in the BIOS interrupt chain. The routine then moves to block 22 which is an interrupt to read the partition sector. Since the protection interrupt handler is at this point actively filtering interrupts in memory, the result of this interrupt via the protection interrupt handler reads the alternative partition sector into memory location 0000:07C0. Control of the PC is then passed to location 0000:07C0 in memory. From here on the partition sector is decrypted and begins executing in conventional fashion under the illusion that it is still, and always has been the first piece of code on the hard disc to be executed.

Fig. 4 shows the operation of the BIOS interrupt chain and the location of th protection interrupt handler.

The BIOS interrupt chain is of conventional design, which is built into the architecture of an Intel processor of the type used in IBM compatible PC's. The interrupt vector table as already discussed contains the addresses of all callable interrupts, and these act as the heads of each linked list chain of interrupts. The last handler to hook a specific interrupt is the first handler notified when such an interrupt occurs. Each individual handler determines whether to hand on notification of an interrupt to its next handler below itself in the interrupt chain. It is for this reason that the protection code saves the vector address before hooking itself into the interrupt chain. Any of the handlers which are hooked into the interrupt chain may return an interrupt directly back to the caller at any stage, thus the protection interrupt handler can wait in memory for any request concerning specific sectors of the hard disc, can pass down to the next adjacent handler any interrupts in which it has no interest, and then deal with those interrupts in which it has an interest. If none of the handlers deal with an interrupt, the last handler passes the interrupt down and the BIOS handler takes control. In general, this is where the real processing of an interrupt occurs, and on the interrupt being processed the BIOS returns the interrupt back to the caller with its resultant value. Block 25 of Fig. 4 starts th

interrupt chain, and block 26 indicates a read write sector request by the operating system. Five interrupt 13h handlers 27 are illustrated namely, handlers Nos. 1 to 5. The interrupt 13h handler No. 3 is the protection interrupt handler. In this case the interrupt 13h handler No. 5 passes the interrupt to the BIOS interrupt 13h handler 28 which returns the interrupt to the caller. Any of the interrupt 13h handlers Nos. 1 to 5 may return the interrupt to the caller.

Turning now to Fig. 5 the operation of the protection interrupt handler No. 3 during normal operation of the PC will now be described. Fig. 5 illustrates a flow chart of the routine which the protection interrupt handler No. 3 executes. Block 30 starts the routine on an interrupt being received by the protection interrupt handler, and the routine moves to block 32. Block 32 determines whether or not the interrupt is a valid authorised installation check by a high level application. If so, the routine moves to block 34 which passes the caller the decryption key for decrypting the alternative partition sector and the alternative boot sector. Block 34 also passes the full address in memory of a flag which indicates whether the protection interrupt handler No. 3 is on or off, and control of the computer is handed to the caller. The

flag is essentially a one byte switch which may be switched off by an authorised caller for maintenance purposes, and when the flag is deemed to be off, all interrupts to the protection interrupt handler No. 3 are ignored and passed on down the interrupt chain to the next adjacent handler. This, thus permits internal maintenance of the system by high level applications. When the flag is on the protection interrupt handler No. 3 operates normally as described.

If block 32 determines that the interrupt is not an installation check, the interrupt should be a valid disc I/O request, and thus the routine moves to block 36, which checks if the flag is on or off. In other words, whether all interrupts are to be dealt with under the control of the protection interrupt handler No. 3. If block 36 determines that the flag has been turned off, then control is passed on down the interrupt handler chain to the next adjacent handler, and the normal disc I/O interrupt eventually passes down to the BIOS handler which reads the hard disc, and then passes the non-decrypted result back to the caller. On block 36 determining that the flag is on, then the routine moves to block 38 which checks whether or not the interrupt is relevant to the hard disc. 13h interrupts are made for both the hard disc, and a floppy disc in either the A or B drive. If block 38

determines that the interrupt is not intended for the hard disc, then the protection interrupt handler No. 3 passes the interrupt down the chain. If however, block 38 determines that the interrupt is intended for the hard disc, the routine moves to block 40 which checks if the interrupt is a write sector interrupt requesting to write to any of the four sectors 1, 2, 3, or 4 illustrated in Fig. 2. If block 40 determines that the interrupt is a write sector interrupt to write to any of the sectors 1 to 4, then the caller is informed that access has been denied. This, thus, ensures that while the protection code is active in memory, none of the four sectors 1 to 4 on the hard disc are written over or corrupted in any way.

Should block 40 determine that the interrupt is not a write sector interrupt, the routine moves to block 44 which checks if the interrupt is a read sector interrupt to read either the standard partition sector 1 or the standard boot sector 2. If block 44 determines that the interrupt is a read sector interrupt to read either the standard partition sector 1 or the standard boot sector 2, then the routine moves to block 46 which sets up an interrupt to the previous handler requesting that the alternative partition sector 3 or the alternative boot sector 4 as the case may be is read. The essential difference here is that

although control passes from one handler to another, an interrupt has been made rath r than a jump, and this requires that a reply be given. Thus, when this interrupt eventually reaches the BIOS handler, and the read is made, it traverses back up the chain from where the call originated, namely, the boot protection handler. At this point, if the original caller requested to read the standard partition sector 1 or the standard boot sector 2, the protection interrupt handler No. 3 calls down to the BIOS interrupt handler requesting the alternative partition sector 3 or the alternative boot sector 4, as the case may be. Since the alternative partition sector 3 and the alternative boot sector 4 are encrypted, the returned sectors are then decrypted by the protection interrupt handler No. 3, and the resultant decrypted sector is passed back up the interrupt chain to the original caller.

Should block 44 determine that the interrupt is not a read interrupt to read the standard partition sector 1 or the standard boot sector 2, the routine moves to block 48 which checks if the interrupt is a read interrupt to read either of the alternative sectors 3 and 4. If so, then the routine moves to block 50 which fills the buffer passed down with the interrupt with zeros, and then passes the buffer back to the caller. This gives the impression to the caller that these two

sectors 3 and 4 are completely blank. Should block 48 determine that the interrupt is not a read sector interrupt for reading the alternative partition and boot sectors 3 and 4, the protection interrupt handler passes the interrupt to the next handler in the chain.

The advantages of the invention are many. By virtue of the fact that the partition table has been moved from the standard partition sector, on booting-up from a system disc in a floppy disc drive, the basic operating system of the PC only sees the protection code when it attempts to access the partition table in the standard partition sector, thereby indicating to the operating system that an invalid message is received, thus indicating that a hard disc is not installed in the computer. Accordingly, the basic operating system is unable to access the hard disc. This information is returned to the user.

However, since the executable protection code is provided in the standard partition sector, initially on the computer being switched on provided there are no floppy discs in the A and B drives, the operating system reads the executable protection code, which then permits boot-up to continue in the normal way from the hard disc, and accordingly, subsequently provides access to the hard disc. As discussed above, should

the hard disc be protected by password protection software, which in general will be the case, access will not be provided to the hard disc after booting-up until the appropriate password has been entered.

CLAIMS

1. A method for preventing access to a hard disc in a computer when the computer has been booted-up from a floppy disc, the hard disc being of the type which would normally have an executable partition code and a partition table at a standard location known as the standard partition sector, the executable partition code being read only prior to booting-up of the computer when the computer is being booted-up from the hard disc, the method comprising the steps of:

transferring the partition code and the partition table to an alternative partition sector on the hard disc, and

writing an executable protection code in the standard partition sector to be read prior to booting-up of the computer when the computer is being booted-up from the hard disc, the protection code comprising;

an instruction for inserting the protection code in the BIOS interrupt chain as a handler for interrupts for access to the standard partition sector, and

an instruction for directing all valid interrupts for access to the standard partition sector of the hard disc to the alternative partition sector.

2. A method as claimed in Claim 1 in which the protection code is insert d in the BIOS interrupt chain just before the BIOS handler.

3. A method as claimed in Claim 1 or 2 in which the partition code and the partition table are encrypted in the alternative partition sector, and the protection code includes a decryption key for decrypting the partition code and the partition table.

4. A method as claimed in any preceding claim in which the boot code and the BPB data code are transferred from a standard boot sector of the hard disc to an alternative boot sector on the hard disc, and the protection code comprises an instruction for directing all valid interrupts for access to the standard boot sector of the hard disc to the alternative boot sector.

5. A method as claimed in Claim 4 in which the boot code and the BPB data code is encrypted in the alternative boot sector.

6. A method as claimed in any preceding claim in which the protection code comprises an instruction for inserting the protection code in memory in the computer prior to inserting the protection handler of the protection code in the BIOS interrupt chain.

7. A method as claimed in any preceding claim in which the protection code comprises an instruction for

ch cking if a read interrupt is received for reading either of the alternative partition or boot sectors, and an instruction to return to the caller a predetermined message unrelated to the code stored in the respective alternative sectors should such a read interrupt be received.

8. A method as claimed in any preceding claim in which the protection code comprises an instruction for checking if a write interrupt is received for writing to either of the alternative partition or boot sectors, and an instruction to return to the caller a message confirming that the respective alternative sectors cannot be written to should such a write instruction be received.

9. A method as claimed in any preceding claim in which the protection code contains the identity of the location of each of the alternative sectors.

10. A method as claimed in any preceding claim for preventing access to a hard disc in a personal computer.

11. A method as claimed in any preceding claim in which the computer is an IBM compatible computer, and the method is for preventing access to the C drive hard

disc of the computer.

12. A method for preventing access to a hard disc in a computer, the method being substantially as described herein with reference to and as illustrated in the accompanying drawings.

13. A computer operating under the control of the method according to any preceding claim for preventing access to a hard disc in the computer after the computer has been booted-up from a floppy disc.

14. A computer comprising a hard disc, the hard disc being of the type which would normally have an executable partition code and an partition table at a standard location known as the standard partition sector, the executable partition code being read only prior to booting-up of the computer when the computer is being booted-up from the hard disc, the hard disc having written thereon at an alternative partition sector the partition code and the partition table, and an executable protection code being written in the standard partition sector to be read prior to booting-up of the computer when the computer is being booted-up from the hard disc, the protection code comprising:

an instruction for inserting the protection code in the BIOS interrupt chain as a handler for interrupts

for access to the standard partition sector, and

an instruction for directing all valid interrupts for access to the standard partition sector of the hard disc to the alternative partition sector.

15. A computer as claimed in Claim 14 in which the instruction for inserting the protection code in the BIOS interrupt chain as a handler for interrupts for access to the standard partition sector is an instruction for inserting the protection code in the BIOS interrupt chain just before the BIOS handler.

16. A computer as claimed in Claim 14 or 15 in which the partition code and the partition table are encrypted in the alternative partition sector, and the protection code includes a decryption key for decrypting the partition code and the partition table.

17. A computer as claimed in any of Claims 14 to 16 in which the boot code and the BPB data code are written in an alternative boot sector on the hard disc which is different to a standard boot sector in which the boot code and the BPB data code are normally written, and the protection code comprises an instruction for directing all valid interrupts for access to the standard boot sector of the hard disc to the alternative boot sector.

18. A computer as claimed in Claim 17 in which the boot code and the BPB data code is encrypted in the alternative boot sector.

19. A computer as claimed in any of Claims 14 to 18 in which the protection code comprises an instruction for inserting the protection code in memory in the computer prior to inserting the protection handler of the protection code in the BIOS interrupt chain.

20. A computer as claimed in any of Claims 14 to 19 in which the protection code comprises an instruction for checking if a read interrupt is received for reading either of the alternative partition or boot sectors, and an instruction to return to the caller a predetermined message unrelated to the code stored in the respective alternative sectors should such a read interrupt be received.

21. A computer as claimed in any of Claims 14 to 20 in which the protection code comprises an instruction for checking if a write interrupt is received for writing to either of the alternative partition or boot sectors, and an instruction to return to the caller a message confirming that the respective alternative sectors cannot be written to should such a write instruction be received.

22. A computer as claimed in any of Claims 14 to 21 in which the protection code contains the identity of the location of each of the alternative sectors.

23. A computer as claimed in any of Claims 14 to 22 in which the computer is a personal computer.

24. A computer as claimed in any of Claims 14 to 23 in which the computer is an IBM compatible personal computer, and the hard disc is the C drive hard disc of the computer.

25. A computer substantially as described herein.

The Patent Office

31

Application No:	GB 9802087.8	**Examiner:**	Mike Davis
Claims searched:	1-25	**Date of search:**	20 July 1998

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.P): G4A (AAP)

Int Cl (Ed.6): G06F

Other:

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
A	US 5012514 (RENTON)	-

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.